

Parallel Processing

Project 5

9 am April 25th

turnin_code: pp_p5

This project is optional, but if you submit one via turnin, it will be graded and averaged with your other project scores. If you do not submit project 5, then the first 4 will be averaged for your final grade.

Enhance project 4 with the following new features.

```
int PGAS_Fadd_init(PGAS_HANDLE handle, int init_val);
```

Creates on a server, a handle for a one-int-sized data item which supports fetch-and-add capability. The int item should be initialized to `init_val`. If more than one of these items is created, they should be round-robin created on the servers with space for load balancing. We can examine the handle to verify this.

```
int PGAS_Fadd(PGAS_HANDLE handle, int incr_val, int *old_val);
```

Using the handle, fetch the `old_val` of the referenced int, and have the value incremented by `incr_val`. Note that `old_val` will obtain the value before the increment. Note that `incr_val` is signed.

init 12

12 fadd - 3 then output is 12+3=15

13 fadd - 5 then output is $13+5 = 18$

```
int PGAS_Locks_init(int num_locks);
```

Collective operation.//collective like broadcast... num_locks

Creates num_locks number of locks on servers so that they are

load balanced, i.e. scattered onto various servers. The locks

can later be referred to by number beginning with 0(0 1,2,3). All are

initialized to un-locked.

```
int PGAS_Lock(int lock_num);//blocking instruction..chances of blocking ..hanging
```

Locks lock_num lock. Hangs until the lock is unlocked if it

is already locked.

```
int PGAS_Unlock(int lock_num);
```

Unlocks lock_num lock, even if it was not locked by this rank.

The makefile should provide a target that will compile (and link) a C

(not C++) program named p5test.c into an executable named p5test.

NOTE that p4test has been changed to p5test in this project.

The user's programs can be run in this way:

```
mpiexec -f some_hosts_filename -n num_ranks ./p5test user_pgm_args
```

Use turnin to submit a tar file containing all of your project, including

a makefile that will build the code. To build, I will type the following:

```
rm -rf *.o
```

```
make
```

Then I will copy some program to p5test.c and type:

```
make p5test
```

and run that program several times.

I may build multiple p5test programs for testing.